

SMAATSDK

**REQUISITOS Y DOCUMENTACIÓN DEL
MÓDULO DE NFC EN ANDROID
RELEASE v1.0**

Serimag

Índice de contenido

Alcance.....	3
Objetivos.....	3
Diagrama de funcionamiento general.....	3
Funciones proporcionadas y parámetros a utilizarlas.....	4
Funciones necesarias para la extracción de datos.....	4
Funciones necesarias para la lectura de los datos extraídos.....	4
Funciones opcionales.....	5
Parámetros a utilizar.....	6
Ejemplo de uso.....	8

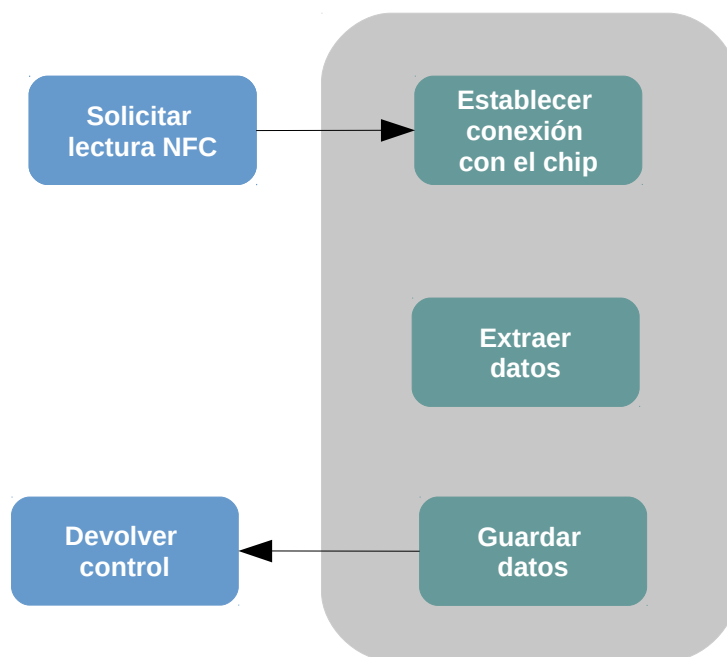
Alcance




Este documento contiene una explicación detallada sobre las funcionalidades y los requisitos del Módulo NFC. En primer lugar se mostrará un diagrama de funcionamiento general. A continuación, se explicarán todas las funciones proporcionadas y sus parámetros. Finalmente, se proporcionará un ejemplo de uso.

Objetivos

Los objetivos de este documento es explicar de manera detallada el funcionamiento y los requisitos del Módulo NFC y facilitar su integración en nuevos proyectos o proyectos ya existentes.

Diagrama de funcionamiento general



-  Bloques propios de la aplicación
-  Bloques encapsulados por el SDK
-  Bloques propios del motor de NFC

Funciones proporcionadas y parámetros a utilizarlas

Funciones necesarias para la extracción de datos

Funciones para extracción de datos	
Estas funciones son necesarias para que el módulo funcione	
NfcInterface nfcInterface = new NfcInterface()	Crear un nuevo objeto NfcInterface
*Añadir funciones opcionales	
void nfcInterface.runId_NFC(Activity activity, int requestCode, String documentId, String birthDate, String expiryDate)	Arrancar la extracción de los datos del DNI mediante NFC. **
void nfcInterface.runPassport_NFC(Activity activity, int requestCode, String documentId, String birthDate, String expiryDate)	Arrancar la extracción de los datos del pasaporte mediante NFC. **
void nfcInterface.runBankCard_NFC(Activity activity, int requestCode)	Arrancar la extracción de los datos de la tarjeta de banco mediante NFC. **
** Activity resultCode	0 → “Keys” de acceso al NFC incorrectos 1 → Extracción correcta de los datos de NFC 2 → Actividad omitida

Funciones necesarias para la lectura de los datos extraídos

Funciones para la lectura de los datos extraídos	
Estas funciones son necesarias para leer los datos extraídos	
JSONObject nfcInterface.getIdData()	Devuelve los datos extraídos del DNI. Dentro del conjunto de datos se diferencia entre: <ul style="list-style-type: none"> • “name” • “surname” • “birthDate” • “gender” • “nationality” • “documentNumber” • “nationalID” (si tiene) • “expiryDate” • “birthPlace” • “address” Excepciones: <ul style="list-style-type: none"> • JSONException

JSONObject nfcInterface.getPassportData()	Devuelve los datos extraídos del pasaporte. Dentro del conjunto de datos se diferencia entre: <ul style="list-style-type: none"> • "name" • "surname" • "birthDate" • "gender" • "nationality" • "documentNumber" • "nationalID" (si tiene) • "expiryDate" Excepciones: <ul style="list-style-type: none"> • JSONException
JSONObject nfcInterface.getBankCardData()	Devuelve los datos extraídos de la tarjeta del banco. Dentro del conjunto de datos se diferencia entre: <ul style="list-style-type: none"> • "PAN" (Permanent Account Number) • "expiryDate" • "cardType"
Bitmap nfcInterface.getPhoto()	Devuelve un bitmap con la foto. Excepciones: <ul style="list-style-type: none"> • Exception

Funciones opcionales

Funciones opcionales	
void nfcInterface.setParameters(boolean photo)	Si se desea extraer la foto del DNI o pasaporte, se tiene que activar setParameters() antes del runId_NFC() o runPassport_NFC() .
void nfcInterface.setWarningTimer(int timerWarning)	Activa mensaje de advertencia en caso de que no se consigue establecer la conexión entre el dispositivo y el chip del documento. Permite parametrizar el timeout que activa el mensaje.
void nfcInterface.setSkipTimer(int timerSkipNFC)	Activa un mensaje de advertencia que no se ha conseguido establecer la conexión entre el dispositivo y el chip del documento y se procede a omitir la lectura, devolviendo el control de la aplicación. Permite parametrizar el timeout que omite la lectura.

Parámetros a utilizar

Parámetros a utilizar	
Parámetros a utilizar en las funciones anteriores	
int intentResultCode	Es el requestCode para obtener el resultado del startActivityForResult() .
String documentId	Este parámetro contiene el número del documento y se utiliza para establecer la conexión con el chip.
String birthDate	Este parámetro contiene la fecha de nacimiento en formato "aammdd" y se utiliza para establecer la conexión con el chip.
String expiryDate	Este parámetro contiene la fecha de caducidad del documento y se usa en formato "aammdd". Junto con los parámetros documentId y birthDate forman la clave de acceso al chip.
boolean photo	Activa la extracción de la foto del DNI.

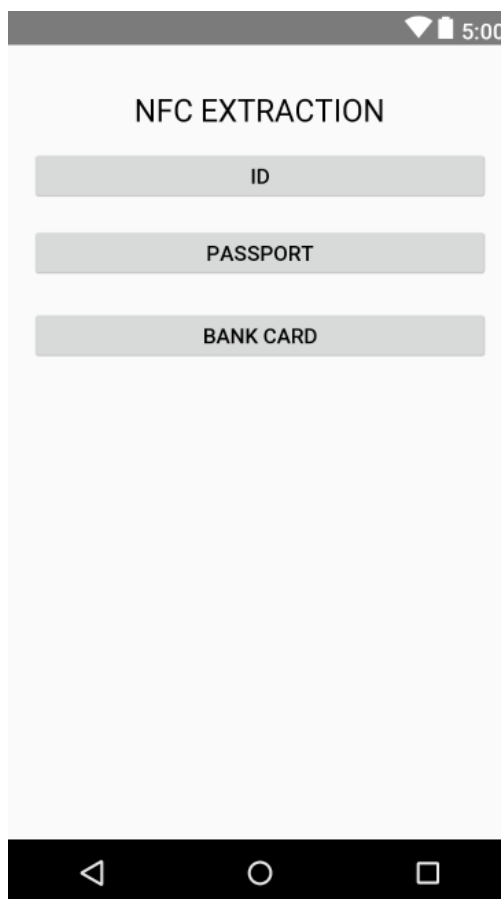
"Keys" de acceso a los JSONs	
"name"	Permite la extracción del NOMBRE del JSONObject que retornan las funciones getIdData() y getPassportData() .
"surname"	Permite la extracción de los APELLIDOS del JSONObject que retornan las funciones getIdData() y getPassportData() .
"birthDate"	Permite la extracción de la FECHA DE NACIMIENTO del JSONObject que retornan las funciones getIdData() y getPassportData() .
"gender"	Permite la extracción del SEXO del JSONObject que retornan las funciones getIdData() y getPassportData() .
"nationality"	Permite la extracción de la NACIONALIDAD del JSONObject que retornan las funciones getIdData() y getPassportData() .
"documentNumber"	Permite la extracción del NÚMERO DEL DOCUMEN TO del JSONObject que retornan las funciones getIdData() y getPassportData() .

“nationalID”	Permite la extracción del Número de Identificación Nacional , en caso de que éste exista, del JSONObject que retornan las funciones getIdData() y getPassportData() .
“expiryDate”	Permite la extracción de la FECHA DE CADUCIDAD del JSONObject que retornan las funciones getIdData() , getPassportData() y getBankCardData() .
“birthPlace”	Permite la extracción del LUGAR DE NACIMIENTO del JSONObject que retornan la función getIdData() .
“address”	Permite la extracción de la Dirección del JSONObject que retornan la función getIdData() .
“PAN”	Permite la extracción del Permanent Account Number (PAN) del JSONObject que retorna la función getIdData() .
“cardType”	Permite la extracción del TIPO de tarjeta bancaria del JSONObject que retorna la función getIdData() .

Ejemplo de uso

Una vez instalado el AAR “nfclibrary.aar”, ya podemos empezar a utilizar el módulo de NFC.

Supongamos que se desea crear una aplicación que extrae los datos y la foto del DNI 3.0 y del pasaporte, además de los datos de la tarjeta de banco mediante NFC. La siguiente pantalla podría ser la pantalla principal:



En esta aplicación de ejemplo tenemos tres botones, uno para cada tipo documental. Una vez extraídos los datos se utiliza la función **onActivityResult()** para mostrar los resultados. En el caso del DNI y del pasaporte los datos se mostrará en una actividad diferente, que en este caso se llama **DataResult**. Para tarjetas de banco, debido a la poca información que se extrae, se mostrará en un textView en la misma actividad, **Main**, o en este caso **StartMenu**. A continuación se mostrará el código de este ejemplo de aplicación básica:


```
01: package serimagmedia.capturaautomatica;
02:
03: import android.content.Intent;
04: import android.os.Bundle;
05: import android.support.v7.app.AppCompatActivity;
06: import android.util.Log;
07: import android.view.View;
08: import android.widget.TextView;
09:
10: import org.json.JSONException;
11: import org.json.JSONObject;
12:
13: import serimagmedia.nfclibrary.NfcInterface;
14:
15:
16:
17: public class StartMenu extends AppCompatActivity {
18:
19:     NfcInterface nfcInterface = new NfcInterface();
20:     TextView textView;
21:     private String docType = "";
22:
23:     @Override
24:     protected void onCreate(Bundle savedInstanceState) {
25:         super.onCreate(savedInstanceState);
26:         setContentView(R.layout.activity_start_menu);
27:
28:         textView = (TextView) findViewById(R.id.idTextView1);
29:
30:     }
31:
32:     private void print(String text) {
33:         Log.d("MAIN", text);
34:     }
35:
36:     public void onClickId(View v){
37:         nfcInterface.setWarningTimer(20);
38:         nfcInterface.setSkipTimer(40);
39:         nfcInterface.setParameters(true);
40:         docType = "ID";
41:         nfcInterface.runId_NFC(this, 7001, "XXXXXXXX", "YYMMDD",
42: "YYMMDD");
43:     }
```

```
45: public void onClickPassport(View v){
46:     nfcInterface.setWarningTimer(20);
47:     nfcInterface.setSkipTimer(40);
48:     nfcInterface.setParameters(true);
49:     docType = "PASSPORT";
50:     nfcInterface.runPassport_NFC(this, 7003, "XXXXXXXX",
51:         "YMMDD", "YMMDD");
52:
53: }
54:
55: public void onClickBankCard(View v){
56:     nfcInterface.setWarningTimer(20);
57:     nfcInterface.setSkipTimer(40);
58:     nfcInterface.runBankCard_NFC(this, 7002);
59: }
60:
61: protected void onActivityResult(int requestCode, int
62:     resultCode, Intent intentData){
63:
64:     if(requestCode == 7001){
65:         if( nfcInterface.getBackButtonPressed() == false){
66:
67:             if(resultCode == 1){
68:                 Intent intent = new Intent(this,
69:                     DataResult.class);
70:                 intent.putExtra("DOCTYPE", docType);
71:                 startActivityForResult(intent, 1);
72:             }else if(resultCode == 0){
73:                 print("KEYS INCORECTAS");
74:             }else if(resultCode == 2){
75:                 print("SKIP ACTIVITY");
76:             }
77:
78:
79:         }
80:     }
```

```
81:         else if(requestCode == 7003) {
82:
83:             if( nfcInterface.getBackButtonPressed() == false){
84:
85:                 if(requestCode == 1){
86:                     Intent intent = new Intent(this,
87:                         DataResult.class);
88:                     intent.putExtra("DOCTYPE", docType);
89:                     startActivityForResult(intent, 1);
90:                 }else if(requestCode == 0){
91:                     print("KEYS INCORRECTAS");
92:                 }else if(requestCode == 2){
93:                     print("SKIP ACTIVITY");
94:                 }
95:             }
96:         }
97:     else if(requestCode == 7002){
98:         if(nfcInterface.getBackButtonPressed() == false){
99:             if(requestCode == 1){
100:                 JSONObject data =
101:                     nfcInterface.getBankCardData();
102:
103:                 if(data != null){
104:                     try {
105:
106: textView.setText("PAN: " + data.getString("PAN") + "\n" +
107:                     "Caducidad: " + data.getString("expiryDate") + "\n" +
108:                     "Tipo de tarjeta: " + data.getString("cardType"));
109:
110:                     } catch (JSONException e) {
111:                         e.printStackTrace();
112:                     }
113:
114:                 }
115:             }else if(requestCode == 2){
116:                 print("SKIP ACTIVITY");
117:             }
118:         }
119:     }
120:
121:
122: }
123: }
```

```
01: package serimagmedia.capturaautomatica;
02:
03: import android.support.v7.app.AppCompatActivity;
04: import android.os.Bundle;
05: import android.view.WindowManager;
06: import android.widget.ImageView;
07: import android.widget.TextView;
08: import org.json.JSONException;
09: import org.json.JSONObject;
10:
11: import serimagmedia.nfclibrary.NfcInterface;
12:
13: public class DataResult extends AppCompatActivity {
14:
15:     NfcInterface nfcInterface = new NfcInterface();
16:     private TextView textView;
17:     private ImageView imageView;
18:     private String doc = "";
19:
20:     @Override
21:     protected void onCreate(Bundle savedInstanceState) {
22:         super.onCreate(savedInstanceState);
23:         setContentView(R.layout.activity_data_result);
24:
25:         //Remove notification bar
26:
27:         this.getWindow().setFlags(WindowManager.LayoutParams.
28: FLAG_FULLSCREEN, WindowManager.LayoutParams.FLAG_FULLSCREEN);
29:
30:         textView = (TextView)findViewById(R.id.passData);
31:         imageView = (ImageView)findViewById(R.id.passportPhoto);
32:
33:         doc = getIntent().getExtras().getString("DOCTYPE");
```

```
34:     if(doc.equals("PASSPORT")){
35:         JSONObject passportData = nfcInterface.getPassportData();
36:         if(passportData != null) {
37:             try {
38:                 textView.setText("Name: " + passportData.get("name") + "\n"
39: +
40:                 "Surname: " + passportData.get("surname") + "\n" +
41:                 "Birth Date: " + passportData.get("birthDate") + "\n" +
42:                 "Gender: " + passportData.get("gender") + "\n" +
43:                 "Nationality: " + passportData.get("nationality") + "\n" +
44:                 "Document Number: " + passportData.get("documentNumber") + "\n"+
45:                 "ID Number: " + passportData.get("nationalId") + "\n" +
46:                 "Expiry Date: " + passportData.get("expiryDate") + "\n");
47:             } catch (JSONException e) {
48:                 e.printStackTrace();
49:             }
50:         }
51:     }else if(doc.equals("ID")){
52:         JSONObject idData = nfcInterface.getIdData();
53:         if(idData != null) {
54:             try {
55:                 textView.setText("Name: " + idData.get("name") + "\n" +
56:                 "Surname: " + idData.get("surname") + "\n" +
57:                 "Birth Date: " + idData.get("birthDate") + "\n" +
58:                 "Gender: " + idData.get("gender") + "\n" +
59:                 "Nationality: " + idData.get("nationality") + "\n" +
60:                 "Document Number: " + idData.get("documentNumber") + "\n" +
61:                 "ID Number: " + idData.get("nationalId") + "\n" +
62:                 "Expiry Date: " + idData.get("expiryDate") + "\n" +
63:
64:                 "Birth Place: " + idData.getString("birthPlace") + "\n"+
65:                 "Address: " + idData.getString("address"));
66:             } catch (JSONException e) {
67:                 e.printStackTrace();
68:             }
69:         }
70:     }
71:     if(nfcInterface.getPhoto() != null){
72:         imageView.setImageBitmap(nfcInterface.getPhoto());
73:     }
74: }
```